

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A system embodied on physical computer readable media that facilitates source code control, comprising:
  - a client-side client control component of a client that tracks an activity associated with a modification of a source code file in a client workspace when the client is in an offline mode, and transmits the activity during an update process when the client moves to an online mode; and
  - a server-side client control component that interfaces to the client-side client control component to facilitate transfer of the activity and update of the source code file in a server repository;

wherein one of the client-side client control component or the server-side client control component checks for an error during the update process and determines whether the update can proceed or must be aborted in part or whole.
2. (Original) The system of claim 1, further comprising:
  - a cache that stores the source code file; and
  - a list that stores the activity, which activity is a command executed during the offline mode and associated with the modification.
3. (Original) The system of claim 2, the source code file stored in the cache remains in an unmodified state.
4. (Original) The system of claim 2, contents of the cache are maintained in both the offline mode and online mode of the client.
5. (Original) The system of claim 1, the client stores all source code files that have been at least one of modified and deleted.

6. (Original) The system of claim 1, the source code file is transmitted to the client before the client moves to the offline mode.

7. (Cancelled)

8. (Original) The system of claim 1, the client stores at least one of the source code file in an unmodified state, pending changeset data, a filetype definition, and a site-specific help file.

9. (Original) The system of claim 1, an error is resolved during a reconciliation process of the activity to the source code file before the source code file can be updated with the modification.

10. (Cancelled)

11. (Previously Presented) The system of claim 1, the source code file is downloaded into the client workspace before the client moves to the offline mode.

12. (Original) The system of claim 1, a pristine copy of the source code file is automatically loaded into a client cache in response to a checkout-related command being executed.

13. (Currently Amended) A system embodied on physical computer readable media that facilitates source code control, comprising:

a cache of a client that stores at least one of a source code file, information relating to activity associated with a modification of the source code file, filetype definitions, and site-specific help files; and

an activity list of the client that stores an activity associated with a modification of the source code file in an arbitrary order which need not be sequential, such that the client checks the activity list for an error during an update process and determines whether to transmit the activity list ~~transmits the activity during an update process~~ when the client moves to an online mode or to abort a part or all of the update process.

14. (Original) The system of claim 13, the source code file stored in the cache remains in an unmodified state, and is a file that has been at least one of modified and deleted in a client workspace.

15. (Original) The system of claim 13, the source code file is stored in the cache of the client during a checkout process that is executed when the client is in one of an offline mode and the online mode.

16. (Original) The system of claim 13, the activity list stores a plurality of activities associated with corresponding commands executed against the source code file, when the client is in an offline mode.

17. (Original) The system of claim 13, the activity is persisted to a server to update a server source code file associated with the source code file during an update process associated with the online mode.

18. (Previously Presented) A computer readable storage medium having stored thereon computer executable instructions for carrying out the system of claim 13.

19. (Original) A computer that employs the system of claim 13.

20. (Original) The system of claim 13, further comprising a classifier that automates a source code control feature by making an inference based on data associated with at least one of the online mode and an offline mode.

21. (Previously Presented) A computer-readable storage medium having computer-executable instructions for performing a method of facilitating source code control, the method comprising:

downloading a copy of a remote source code file from a remote location to a client;

storing activity data on the client during an offline mode, which activity data is associated with a modification to the copy;

performing an error check to determine if a conflict exists, if the remote source code file is locked, if an administrator has disallowed an update process;

determining whether an update of at least one of a portion and all of the activity data can proceed if an error is detected during the update process or if the error is unresolvable and the update cannot proceed; and

transmitting the activity data to the remote location during an online mode to update the remote source code file during an update process if no unresolvable error is detected.

22. (Original) The method of claim 21, further comprising executing one or more commands on the client during the offline mode to facilitate caching of a pristine version of the copy.

23. (Original) The method of claim 21, further comprising caching a pristine version of the copy in response to making the modification to the source code file on the client.

24. (Cancelled)

25. (Original) The method of claim 21, further comprising resolving a conflict between a command executed during modification of the copy and the remote source code file during the update process.

26. (Original) The method of claim 21, the act of storing occurs in response to execution of a checkout-related command on the client.

27. (Original) The method of claim 21, further comprising transmitting a modified source code file to the remote location after the activity data has been reconciled to the remote source code file.

28. (Currently Amended) A method of facilitating source code control, comprising:  
moving a copy of a remote source code file of a server to a client workspace of a client;

    caching the copy in a client cache in response to a modification of the copy,  
which modification is one of a plurality of modifications;

    storing activity data on the client during an offline mode, which activity data is associated with the plurality of modifications made to the copy;

    performing an error check to determine if a conflict exists, if the remote source code file is locked, if an administrator has disallowed an update process;

determining whether an update of at least one of a portion and all of the activity data can proceed if an error is detected during the update process or if the error is unresolvable and the update cannot proceed; and

    transmitting the activity data to the server during an online mode to update the remote source code file during an update process if no unresolvable error is detected.

29. (Original) The method of claim 28, further comprising storing information at the client before entering the offline mode.

30. (Original) The method of claim 28, further comprising updating a checkout record at the server during the online mode.

31. (Original) The method of claim 28, further comprising issuing a command to enter the offline mode, and a corresponding command to enter the online mode.

32. (Original) The method of claim 28, further comprising issuing a command that includes a URL to a workspace.

33. (Original) The method of claim 28, further comprising at least one of:  
checking out the copy during the act of moving, which is during the online mode;  
and  
checking out the copy from a client workspace during the offline mode.

34. (Currently Amended) The method of claim 28, further comprising:  
detecting an error during the offline mode;  
presenting an error message associated with the error;  
maintaining the client in the offline mode in response to detecting the error; and  
allowing the client to move to the online mode after the error has been resolved.

35. (Original) The method of claim 28, further comprising imposing permissions required for the offline mode, during the online mode.

36. (Original) The method of claim 28, further comprising caching at the client at least one of unmodified files, pending changeset information, filetype definitions, and site-specific help files.

37. (Original) The method of claim 28, further comprising reapplying a checkout process to the server when at least one of the checkout was cancelled at the server when the client was offline and a checkout was performed offline on the client after the copy was downloaded to the client without the checkout process issued to the server during the online mode.

38. (Previously Presented) A system that facilitates source code control, comprising:
- means for moving a copy of a remote source code file of a server to a client workspace of a client;
  - means for storing at the client at least one of an unmodified version of the copy, pending changeset information, a filetype definition, and a site-specific help file;
  - means for caching the unmodified version of the copy in response to a checkout process of the copy;
  - means for storing command data of the client during the offline mode, which command data is associated with the plurality of modifications to the copy;
  - means for performing an error check to determine if a conflict exists, if the remote source code file is locked, if an administrator has disallowed an update process;
  - means for determining whether an update of at least one of a portion and all of the activity data can proceed if an error is detected during the update process or if the error is unresolveable and the update cannot proceed; and
  - means for transmitting the command data to the server during an online mode to update the remote source code file during an update process if no unresolveable error is determined.
39. (Original) The system of claim 38, means for imposing all permissions required for the offline mode, during the online mode.
40. (Previously Presented) The system of claim 38, further comprising means for resolving a detected error during reconciliation of the command data to the server, before uploading the modified version of the copy to the server.

41. (Previously Presented) A system that facilitates source code control, comprising:

a client-side client control component of a client that tracks an activity associated with a modification of a source code file in a client workspace when the client is in an offline mode; ~~checks for errors~~; and transmits the activity to be performed during an update process when the client moves to an online mode; ~~further checking for an error or a conflict during the update process~~;

a server-side client control component that interfaces to the client-side client control component to facilitate transfer of the activity and update of the source code file in a server repository;

wherein one of the client-side client control component and the server-side client control component checks for an error during the update process and determines whether the update can proceed or must be aborted in part or whole;

a cache that stores the source code file whether the user is in an online mode or an offline mode, the source code file downloaded to a client workspace and stored in the cache remains in an unmodified state;

the cache further storing information relating to activity associated with a modification of the source code file, filetype definitions, and site-specific help files; and

a list that stores the activity, which activity is a command executed during the offline mode and associated with the modification.